WHAT IS CLAIMED IS:

1. A method for peer-to-peer messaging between network resources comprising:

communicating with a first process by writing a first text file in a first scratch space, where the first text file describes at least one of at least a first set of information that a second process has generated and at least a first action to be performed on the first set of information;

detecting, by a first arbiter, the first text file, wherein the first arbiter is implemented by the first process; and

performing at least one of:

implementing, by the first arbiter, the first action; and

applying, by the first arbiter, logic embedded within the first arbiter to determine actions to be performed on the first text file.

2. A method in accordance with Claim 1 wherein communicating with the first process comprises communicating with the first process by writing an American standard code for information exchange (ASCII) file.

3. A method in accordance with Claim 2 wherein communicating with the first process by writing the ASCII file comprises communicating with the first process by writing one of a hypertext markup language (HTML) file, an extensible HTML (XML) file, a multipurpose internet mail extensions (MIME) file, a .NET file, and a simple object access protocol (SOAP) file in the first scratch space

4. A method in accordance with Claim 1 wherein applying, by the first arbiter, logic embedded within the first arbiter comprises at least one of:

moving the first text file to a second scratch space;

moving the first set of information to the second scratch space; and

obtaining index information from the first text file and moving images associated with the index information into a storage repository.

5. A method in accordance with Claim 1 further comprising encoding the first set of information within the first text file.

6. A method in accordance with Claim 1 further comprising referencing the first set of information as being in an external file.

7. A method in accordance with Claim 1 further comprising:

enabling, by the first process, an input from a user; and

writing the input to at least one of the first text file and a second text file in the first scratch space.

8. A method in accordance with Claim 1 wherein communicating with the first process comprises communicating with an image display process by writing the first text file in the first scratch space.

9. A method in accordance with Claim 1 wherein communicating with the first process comprises communicating with the first process by writing the first text file in the first scratch space, wherein the first text file describes at least one of an image that a scanning process has generated and the first action to be performed on the image.

10. A method in accordance with Claim 1 further comprising:

reading, by the first arbiter, instructions within the first text file.

11. A method in accordance with Claim 1 wherein applying, by the first arbiter, logic embedded within the first arbiter comprises determining whether data that is referenced by the first text file as being in a second text file should be processed.

12. A method in accordance with Claim 1 further comprising:

communicating with a third process by writing a second text file in a second scratch space, wherein the second text file describes at least one of a second set of information that the first process has displayed and a second action to be performed on the second set of information;

detecting, by a second arbiter, the second text file, wherein the second arbiter is implemented by the third process; and

performing at least one of:

implementing, by the second arbiter, the second action; and

applying, by the second arbiter, logic embedded within the second arbiter to determine actions to be performed on the second text file.

13. A method in accordance with Claim 1 further comprising:

specifying a format of the first text file; and

changing the format of the first text file to the specified format.

14. A method in accordance with Claim 13 where changing the format of the first text file includes one of:

converting the first text file from a plain text file to a hypertext markup language (HTML) file; and

converting the first text file from a simple object access protocol (SOAP) to a .NET file; and

restructuring data within the first text file.

15. A method in accordance with Claim 1 further comprising:

requesting a public key from an authority;

encrypting a portion of the first text file by using the public key;

signing the portion;

transmitting the portion and the public key to a second scratch space; and

requesting an authentication of a second process that received the portion and the public key.

16. A method in accordance with Claim 15 further comprising:

requesting an authentication of the digital signature;

further transmitting the portion from the second process to a service on obtaining the authentication of the second signature;

decrypting the portion using a private key; and

sending the decrypted portion from the service to the second process.

17. A method in accordance with Claim 1 further comprising applying, by the first arbiter, at least one of a File Transfer Protocol (FTP), a Hypertext Transfer Protocol (HTTP), and a file services network protocol to move the first text file between network resources.

18. A method for peer-to-peer messaging between network resources comprising:

reading a text file received within a scratch space to find a digital signature within the text file;

comparing the digital signature to the contents of the text file to determine whether the digital signature is valid;

moving the text file to a queue designated for improper files on determining that the digital signature is invalid; and

processing the text file on determining that the digital signature is valid.

19. A method in accordance with Claim 18 further comprising:

comparing the digital signature with signatures in a list in a database on determining that the digital signature is valid, wherein the comparison is made to determine whether there is permission to transmit the text file.

20. A method in accordance with Claim 18 wherein processing the text file comprises processing according to one of an independent rule set of an arbiter that reads the text file and instructions within the text file.

21. A method for peer-to-peer messaging between network resources comprising:

reading a text file to find at least one portion of the text file and to find a digital signature within the portion, wherein the portion is designated for processing;

comparing the digital signature to the contents of the portion of the text file to determine whether the digital signature is valid;

moving the portion of the text file to a queue designated for improper portions of files on determining that the digital signature is invalid; and

processing the portion of the text file on determining that the digital signature is valid.

22. A method in accordance with Claim 21 further comprising:

comparing the digital signature with signatures in a list in a database on determining that the digital signature is valid, wherein the comparison is made to determine whether there is permission to transmit the portion of the text file.

23. A method in accordance with Claim 21 wherein processing the portion of the text file comprises processing according to one of an independent rule

set of an arbiter that reads the text file and instructions within the portion of the text file.

24. A method for peer-to-peer messaging between network resources comprising:

obtaining, by a first arbiter, a first output of a first process;

parsing information within the first output into a first set of text files;

writing the first set of text files into a first text file in at least one of a first scratch space and a second scratch space;

detecting, by a second arbiter of a second process, the first text file;

reading, by the second arbiter, the first text file; and

performing an independent operation on the first text file based on rules in the second arbiter to obtain a second output.

25. A method in accordance with Claim 24 wherein performing the independent operation comprises referring to at least one of external databases, external data files, and external objects to perform the independent operation.

26. A method in accordance with Claim 24 further comprising writing the first output to at least one of the first scratch space and other scratch spaces.

27. A method in accordance with Claim 24 further comprising:

obtaining, by a third process, the second output;

parsing information within the second output into a second set of text files;

writing at least one of the files in the second set into a third text file in a third scratch space;

detecting, by a third arbiter, the third text file;

reading, by the third arbiter, the third text file; and

performing an independent operation on the third text file based on rules in the third arbiter to obtain a third output.

28. A method in accordance with Claim 24 further comprising formatting, by the first arbiter, information available to the first arbiter.

29. A method in accordance with Claim 28 further comprising sending, by the first arbiter, the information to the second process via an interface.

30. A method in accordance with Claim 29 wherein sending, by the first arbiter, the information comprises sending, by the first arbiter, an image to display the image along with a document of the second process.

31. A method in accordance with Claim 29 wherein sending, by the first arbiter, the information comprises sending, by the first arbiter, an image of a check to display the image along with a document of the second process.

32. A network system comprising:

an originating arbiter configured to communicate with a first process by writing a first text file in a first scratch space, wherein the first text file describes at least one of at least a first set of information that a second process has generated and at least a first action to be performed on the first set of information; and

a first arbiter implemented by the first process, the first arbiter configured to:

detect the first text file; and

perform at least one of:

implementation of the first action; and

application of logic embedded within the first arbiter to determine actions to be performed on the first text file.

33. A network system comprising:

an arbiter configured to:

read a text file received within a scratch space to find the digital signature within the text file;

compare the digital signature to the contents of the text file to determine whether the digital signature is valid;

move the text file to a queue designated for improper files on determining that the digital signature is invalid; and

process the text file on determining that the digital signature is valid.

34. A network system comprising:

an arbiter configured to:

read a text file to find at least one portion of the text file and to find a digital signature within the portion, wherein the portion is designated for processing;

compare the digital signature to the contents of the portion of the text file to determine whether the digital signature is valid;

move the portion of the text file to a queue designated for improper portions of files on determining that the digital signature is invalid; and

process the portion of the text file on determining that the digital signature is valid.

35. A network system comprising:

a first arbiter configured to:

obtain a first output of a first process;

parse information within the first output into a first set of text files; and

write the first set of text files into a first text file in at least one of a first scratch space and a second scratch space; and

a second arbiter configured to:

detect the first text file;

read the first text file; and

perform an independent operation on the first text file based on rules in the second arbiter to obtain a second output.